Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

# Aprendizagem de Língua Assistida por Computador: Uma Abordagem Baseada em HPSG

Flávio Maico Vaz da Costa

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação – Licenciatura

Orientador
Prof. Dr. José Carlos Loureiro Ralha

Brasília
2005

Universidade de Brasília – UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Curso de Computação – Licenciatura

Coordenador: Prof. Dr. Marco Aurélio de Carvalho

Banca examinadora composta por:

Prof. Dr. José Carlos Loureiro Ralha (Orientador) – CIC/UnB
Prof.ª Dr.ª Fátima Bernardes – CIC/UnB
Prof. Dr. Fernando Pereira – AT&T Research Group
Prof. Dr. Noam Chomsky – Linguistics/MIT

**CIP – Catalogação Internacional na Publicação**

Flávio Maico Vaz da Costa.

Aprendizagem de Língua Assistida por Computador: Uma Abordagem Baseada em HPSG/ Flávio Maico Vaz da Costa. Brasília : UnB, 2005.
43 p. : il. ; 29,5 cm.

Monografia (Graduação) – Universidade de Brasília, Brasília, 2005.

1. verificação sintática, 2. aprendizagem de língua, 3. HPSG, 4. LKB

CDU 004

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro – Asa Norte
CEP 70910–900
Brasília – DF – Brasil

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

# Aprendizagem de Língua Assistida por Computador: Uma Abordagem Baseada em HPSG

Flávio Maico Vaz da Costa

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação – Licenciatura

Prof. Dr. José Carlos Loureiro Ralha (Orientador)
CIC/UnB

Profª. Drª. Fátima Bernardes
CIC/UnB

Prof. Dr. Fernando Pereira
AT&T Research Group

Prof. Dr. Marco Aurélio de Carvalho
Coordenador do Curso de Computação – Licenciatura

Brasília, 20 de junho de 2005

# *Agradecimentos*

Exemplo tosco: agradeço a todos que me auxiliaram nessa loucura universitária, como o Prof. Dr. Fernando Pereira e seu Prolog e o Prof. Dr. Noam Chomsky com suas gramáticas formais.

# *Resumo*

Este trabalho analisa fundamentos e métodos pertinentes para o planejamento e desenvolvimento de sistemas de processamento de linguagem natural cujo objetivo é auxiliar a aprendizagem de uma língua, sobretudo como estudante estrangeiro. O foco principal é a correção sintática baseada no formalismo HPSG conforme implementado no sistema LKB (*Linguistic Knowledge Building*). Foi desenvolvido um analisador em Java que, juntamente com uma gramática da língua espanhola de médio porte, permite experimentar a técnica de correção através de regras de malformação.

**Palavras-chave:** verificação sintática, aprendizagem de língua, HPSG, LKB

# *Abstract*

Here will go a translation, when the Portuguese version is OK.

**Keywords:** syntax checking, language learning, HPSG, LKB

# *Sumário*

# *Lista de Figuras*

# *Lista de Tabelas*

# Glossário

**Análise profunda**

Mais informação gramatical. 25

**Análise superficial**

Uma análise bem de leve. 25

**Núcleo**

Constituinte principal. 32

**Regra imanente**

Descritiva. 18

**Regra transcendente**

Normativa. 18

**Suavização (*smoothing*)**

Atenua problemas com dados esparsos. 25

**Traço (*feature*)**

Falar aqui da dimensão linguística. Não confundir com vestígio da gramática transformacional (*trace*). 28

# Capítulo 1

# Introdução

Processamento de Linguagem Natural (PLN) é um campo de estudos multidisciplinar, integrando Lingüística, Matemática, Computação, dentre outras disciplinas afins. Sua aplicação em sistemas de cunho educacional diversifica ainda mais o conhecimento envolvido, particularmente de Ciências Humanas tais como a Pedagogia, a Psicologia, a Sociologia. Mesmo considerando os progressos já alcançados nessa área, a complexidade envolvida nela continua sendo um fator oneroso, dificultando a implantação de aplicações tão robustas, precisas e abrangentes quanto desejável.

Nesse cenário é imperioso responder à questão: vale à pena investir em pesquisa e desenvolvimento nessa área? Para obtermos uma resposta satisfatória, podemos observar o uso da informática como instrumento de aprendizagem de línguas, e a contribuição que o PLN pode oferecer.

Os sistemas de apoio à aprendizagem de língua mais simples restringem-se a questões de múltipla escolha e preenchimento de lacunas. Com um pouco mais de sofisticação pode-se criar atividades mais interativas e variadas, seja com recursos simples de "arrastar-e-soltar" até pequenos jogos multimídia. Tais abordagens são relativamente fáceis de elaborar e podem ser organizadas de forma que o sistema possa determinar a resposta certa de maneira computacionalmente eficiente e inequívoca. Na literatura em inglês, sistemas como esses são categorizados sob a sigla CALL — *Computer-Aided Language Learning*. Poderíamos empregar a sigla em português ALAC — Aprendizagem de Língua Assistida por Computador, mas será utilizada aqui sua equivalente em inglês, forma na qual está consagrada.

Incorporando processamento de linguagem natural a sistemas de CALL, abre-se uma nova dimensão: a da criatividade. Ao invés de simplesmente

escolher entre opções pré-determinadas, pode-se oferecer ao aluno questões de autêntica produção de texto. Vejamos a seguinte questão, retirada da prova de Língua Portuguesa e Literaturas de Língua Portuguesa do exame Vestibular da Unicamp (1997)[1]:

> *Texto: "PF prende acusado de terrorismo nos EUA*
>
> *O libanês Marwán Al Safadi, suspeito do atentado ocorrido no World Trade Center em Nova York (EUA), em 1993, foi preso no último dia 6 em Assunção (Paraguai), após ser localizado pela PF (Polícia Federal) [. . . ]"*
>
> *Pergunta: "A que fato mencionado no título refere-se a expressão 'nos EUA', considerando o sentido geral da notícia?"*

Um sistema tradicional poderia oferecer algumas opções, por exemplo:

1. Ao local da prisão do suspeito;
2. Ao local do ato terrorista praticado;
3. Ao local da acusação de terrorismo;
4. Nenhuma das alternativas.

Podemos perceber que nesse tipo de questão o uso de opções pré-definidas não seria satisfatório, pois restringiria artificialmente o exercício de interpretação de texto. Em situações corriqueiras, a análise de um texto geralmente é feita sem qualquer conhecimento sobre as possíveis interpretações. É desejável, portanto, que a questão seja respondida através de uma ou duas sentenças livres. Se o sistema for capaz de realizar análise sintática, pode identificar eventuais erros de sintaxe e, se for necessário, explicar a regra gramatical sendo violada. Se dispuser de análise semântica adequada, pode inclusive verificar se a interpretação do aluno condiz com a esperada.

Tutores de línguas capazes de receber entrada em linguagem natural enquadram-se na categoria de ICALL — *Intelligent Computer-Aided Language Learning*. Os sistemas de ICALL podem ser considerados como um tipo específico de sistemas de CALL. O exemplo acima serve como uma ilustração simples das possibilidades de PLN como um recurso enriquecedor de sistemas de aprendizagem. Nos capítulos seguintes, tais possibilidades serão exploradas em maior detalhe.

[1] Comvest, Pró-Reitoria de Graduação: http://www.comvest.unicamp.br/

Reconhecidos os benefícios dos tutores inteligentes de línguas, resta outro questionamento importante: devido à dificuldade do tratamento computacional de linguagem natural, seu uso em aplicações realistas é viável? Atualmente, nenhum sistema é capaz de reconhecer com precisão a totalidade de uma língua natural. Entretanto, essa afirmação deve ser feita com a ressalva que, devido à ficção da homogeneidade (Lyons, 1981, p. 35-37), sequer podemos definir com precisão o que seria "a totalidade de uma língua". Tais preocupações são contornadas com o entendimento que *perfeição* não é requisito para *utilidade*.

Por outro lado, os desafios a serem enfrentados na construção de um sistema que atenda aos requisitos de qualidade almejados ainda são muitos, embora a pesquisa na área se mantenha buscando novos paradigmas e estratégias para, com maior facilidade, resolver os obstáculos atuais. As teorias lingüístico-computacionais de que agora dispomos são muito mais satisfatórias do que aquelas de algumas décadas atrás, as quais se baseavam essencialmente em autômatos e gramáticas livres de contexto. Pesquisas recentes estão constantemente propondo novos formalismos mais precisos, completos e expressivos.

A *Head-Driven Phrase Structure Grammar* (HPSG) é uma dessas teorias contemporâneas que tem sido objeto ativo de pesquisa e foi empregada com sucesso em vários sistemas computacionais que são capazes, com diferentes graus de sucesso, utilizar linguagem natural (Makino et al., 1997; Copestake & Flickinger, 2000; Goyal et al., 2003).

Os capítulos seguintes desenvolverão: as teorias e fundamentos das gramáticas formais e aplicações em ICALL, uma exposição sobre a implementação computacional dessas teorias, apresentação de um protótipo para experimentar correção sintática com HPSG, uma digressão sobre os resultados obtidos e conclusões.

# Capítulo 2

# Considerações sobre aprendizagem

## 2.1 Competências lingüísticas

Para desenvolver uma aplicação de computação aplicada à educação deve-se observar as competências a serem desenvolvidas pelo aluno. Considerando especificamente a aprendizagem de línguas, tais competências enquadram-se em quatro grupos básicos: fala, escrita, leitura e escuta.

| | Produção | Compreensão |
|---|---|---|
| Textual | Escrita | Leitura |
| Oral | Fala | Escuta |

Tabela 2.1: Competências lingüísticas básicas

Processamento de linguagem natural é particularmente útil como parte de sistemas que trabalham as competências de produção, ou seja, a escrita e a fala. Ainda que ambos os casos possam ter como base algoritmos semelhantes, o processamento de interação oral requer um módulo adicional para reconhecimento de fala, aumentando sobremaneira a complexidade da solução a ser desenvolvida. Devido a este motivo, discutiremos apenas a produção textual, isto é a escrita.

Por outro lado, também é possível trabalhar as competências de compreensão sem maiores dificuldades. A interpretação de texto é uma maneira natural de instigar a capacidade de leitura e análise do aluno. A escuta pode ser treinada através de trechos de rádio, entrevistas ou diálogos especificamente criados para fins pedagógicos. Enunciados de questões, ao invés de escritos, podem ser narrados por mensagens previamente gravadas ou por síntese de voz, uma tecnologia que já está madura o suficiente

para sair do meio acadêmico para aplicações comerciais bem sucedidas[1].

Há pelo menos duas outras variáveis críticas a serem avaliadas na construção de qualquer sistema de ensino de línguas: o perfil dos usuários aos quais se destina e a língua que se pretende ensinar. Quanto aos usuários:

1. São crianças, jovens ou adultos?
2. Têm aprendizagem normal ou apresentam algum distúrbio, tal como a dislexia?
3. São portadores de algum tipo de necessidade especial, devido à surdez, cegueira ou outros?

Quanto à língua a ser ensinada:

1. É sua língua nativa, segunda língua ou língua estrangeira?
2. Qual o seu nível de proficiência quanto à língua, básico, intermediário, avançado?
3. Quais são as similaridades e diferenças entre sua língua nativa e a língua sendo aprendida?

Os critérios envolvidos em ambas as variáveis eventualmente interagem de maneira não evidente à primeira vista. Por exemplo, McCoy & Masterman (1997) afirmam:

> "[...]nós queremos enfatizar a opinião de que o inglês é, para nativos em ASL [linguagem americana de sinais], uma língua fundamentalmente diferente e desafiadora, motivando a necessidade de adotar uma estratégia de Aquisição de Segunda Língua para facilitar o processo de aprendizagem." [Todas as traduções são minhas, devo informar isso? Como?]

Ainda sobre esta questão, adiante enfatizam as dificuldades peculiares ao estudante surdo:

[1]Como um exemplo atual, os personagens desenvolvidos pela Oddcast (http://www.oddcast.com/) que, segundo informação veiculada na página principal da empresa, suportam 64 línguas, dentre elas o português.

> "Devemos notar que, enquanto há 'verificadores de estilo' e 'verificadores gramaticais' no mercado, esses programas não satisfazem as necessidades do surdo. Educadores de surdos (e outras pessoas trabalhando com indivíduos surdos) relatam que tais verificadores, voltados para os erros daqueles que escrevem tendo capacidade auditiva, frustram estudantes surdos. Adaptados para o estilo de escrita de falantes fluentes, nativos de inglês, eles não capturam vários erros que são comuns na escrita de surdos [...]."

Outro exemplo que destaca a interação entre esses critérios diz respeito ao processo de desenvolvimento da escrita na criança. Mayher et al. (1983) descrevem esse processo pelo trinômio *fluência, clareza e correção*, defendendo que estes três acontecem simultaneamente, mas que o aluno define a ênfase que dará a cada um desses aspectos - os quais também podem ser observados, talvez de forma até mais acentuada, em estudantes de uma segunda língua. Em seguida, afirmam que uma significativa parcela dos estadunidenses coloca a *correção* em primeiro lugar. Ainda mais importante é quando declaram que "pesquisas indicam que a única forma pela qual se aprende a escrever é escrevendo". Encontramos aqui uma contribuição significativa que o PLN pode oferecer para uma aprendizagem de línguas através do computador mais efetiva.

## 2.2 Distinção entre aquisição e aprendizagem

A teoria consagrada por Krashen (1981) sobre a *aquisição* de uma segunda língua, em contraste com a *aprendizagem*, levanta questões pertinentes sobre sistemas de ICALL: qual o papel de um sistema de correção gramatical no desenvolvimento da competência lingüística em uma segunda ou terceira língua? Que características um sistema desses deve apresentar?

Warschauer (1996) enquadra os sistemas de CALL em três fases ao longo das quais as respostas para esses questionamentos se tornaram mais claras:

1. Behaviorista, quando a exposição continuada ao material pedagógico e repetição de exercícios informatizados num formato "pergunta/alternativa

correta" bastante rígido eram consideradas técnicas adequadas, ou mesmo essenciais, na aprendizagem de uma língua.

2. Comunicativa, com o foco na criação de exercícios mais interativos e criativos que os predecessores. Outra mudança foi um maior esforço em integrar o computador à vivência da sala de aula, encarando-o como uma ferramenta para encorajar a discussão entre os alunos, a escrita, o pensamento crítico; cada atividade com um propósito bem definido.

3. Integrada, contando com o desenvolvimento das tecnologias multimídia, da Internet e a popularização do computador hoje podemos ter aplicativos que permitem o desenvolvimento simultâneo de várias competências lingüísticas. Mesmo com esse avanço, os programas atuais não têm inteligência suficiente para *interagir* satisfatoriamente com o estudante. A Inteligência Artificial desponta como futura solução para essa deficiência.

Enquanto a transição para a terceira fase foi predominantemente tecnológica, provavelmente o maior progresso teórico-metodológico se deu na segunda fase. Underwood (1984, apud Warschauer, 1996) propôs as seguintes premissas como desejáveis em um sistema de aprendizagem de língua:

- Focar no uso das formas [lingüísticas] e não nas formas em si;
- Ensinar gramática preferencialmente de maneira implícita ao invés de explicita;
- Permitir e encorajar que os estudantes criem sentenças originais, não apenas manipular linguagem pré-fabricada;
- Não julgar e avaliar tudo que o aluno faz nem recompensa-lo com mensagens de congratulação, luzes e sirenes;
- Evitar dizer aos estudantes que estão errados e ser flexível nas respostas que aceita;
- Usar exclusivamente a língua-alvo e criar um ambiente no qual seu uso pareça natural, tanto na tela quanto fora dela;
- Nunca tentar fazer algo que um livro possa fazer igualmente bem.

Finalmente, Warschauer (ibid.) propõe uma classificação de aplicações de CALL na qual afirma que “eles (verificadores gramaticais) geralmente são bastante confusos para estudantes de uma língua estrangeira” e que são indicados para falantes nativos. Embora haja alguma razão nessa ressalva, aparentemente ela é motivada mais pelo uso inapropriado da tecnologia que por uma inadequação intrínseca desta tecnologia como suporte ao ensino de uma segunda língua.

A Tabela 2.2 relaciona certas distinções teóricas e as tendências de desenvolvimento das competências lingüísticas sob a luz da hipótese de *monitores* de Krashen (1981). Pode-se associar a *aquisição* à idéia de *competência* introduzida por Chomsky, à de *parole* de Saussure, onde ocorre naturalmente a assimilação de regras imanentes (Lyons, 1981, p. 54–55) . A *aprendizagem* pode ser compreendida, respectivamente, como o estudo formal de um *sistema lingüístico*, ou *langue*, ou ainda a assimilação de regras transcendentes (ibid.) .

| | Subconsciente | Consciente |
|---|---|---|
| Chomsky | Competência lingüística | Sistema lingüístico |
| Saussure | *Parole* | *Langue* |
| Regras | Imanentes | Transcendentes |
| Acentuado em | Crianças até doze anos | Nativos de outra língua com mais de doze anos |

Tabela 2.2: Assimilação de competências lingüísticas

Krashen defende que a aprendizagem só ocorre como um monitor, isto é, através de um esforço consciente. A aquisição seria o processo de internalização de uma língua, subconsciente e particularmente notável em crianças até os doze anos de idade:

> “A Teoria dos Monitores, e suas propostas inter-relações com aptidão e atitude, mostram uma imagem mais clara da causa das diferenças entre crianças e adultos no desenvolvimento de uma segunda língua. Eu sugeri (Krashen, 1975a) que a origem do Monitor são as *operações formais*, um estágio que muitas pessoas, mas não todas, alcançam por volta dos 12 anos (Inhelder & Piaget, 1958). O pensador formal tem a capacidade de ’manipular... verbalmente as relações entre idéias na ausência de proposições empíricas anteriores ou concorrentes’ (Ausubel & Ausubel, 1971,

> p. 63). Para pensadores formais, novos conceitos são adquiridos primariamente a partir de 'experiências verbais ao invés de concretas' (ibid., p. 66). O pensador formal também tem uma meta-percepção de suas idéias e pode usar *regras* abstratas para resolver toda uma classe de problemas de uma só vez. Assim é plausível que a capacidade de usar uma gramática consciente, que requer uma meta-percepção da linguagem e regras abstratas gerais, surge como resultado de operações formais." [grifo do autor]

Visto que a capacidade analítica de usar as regras de maneira produtiva é dependente de vários fatores, como idade, familiaridade com a língua, ser ou não a língua nativa do estudante, até mesmo traços de personalidade (Krashen, 1981), evidentemente a abordagem adotada num sistema de CALL deve se adequar ao perfil do aprendiz, usando esses fatores como parâmetros. Enquanto para falantes nativos adultos e estudantes avançados o contato com as regras gramaticais pode ser mais explícito e formal, nos demais casos o ideal é que o programa simule com a maior fidelidade possível uma vivência concreta da língua-alvo, empregando recursos como a síntese de voz. Contudo, tal objetivo está além das atuais possibilidades de sistemas computacionais, pois envolveria o emprego de tecnologias ainda insuficientemente desenvolvidas, como mecanismos eficientes e abrangentes de representação semântica para uma interação mais realista com o estudante (Warschauer & Healey, 1998).

A despeito dessas considerações, como será apresentado adiante, existem casos bem sucedidos de aplicações com regras gramaticais explícitas voltadas para estudantes iniciantes.

## 2.3 Zona de desenvolvimento proximal

Outra dimensão psicolingüística pertinente surge pela análise segundo a zona de desenvolvimento proximal (ZDP), de Vygotsky. Tharp & Gallimore (1988) interpretaram o desempenho de uma determinada competência em quatro estágios gerais, ilustrados na Figura 2.1 (Adaptada da mesma obra, p. 35. O problema é que eu não li esse livro, mas ela é quase um clássico e muita gente faz referência. Eu não achei esse gráfico em nenhum texto acadêmico, apenas em sites da internet... o que eu faço, finjo

que eu li o livro ou tento dar um jeito de conseguí-lo?):

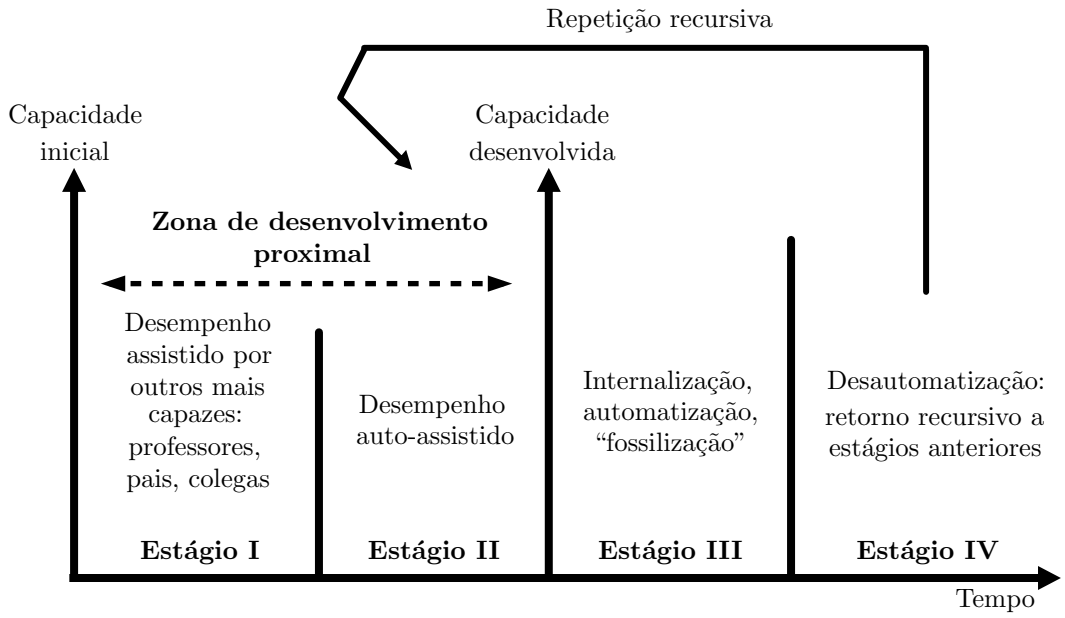


Figura 2.1: Quatro estágios da ZDP

1. Desempenho assistido por outros mais capazes: o desempenho só é possível com a assistência de terceiros mais capacitados, é o estágio inicial da aprendizagem quando ainda não há autonomia. A idade e o tipo de tarefa determinam o auxílio que é necessário. Aqui ocorre o que Vygotsky chama de "imitação", não uma imitação mecânica, mas com cada vez mais entendimento. A interação com os outros e o feedback recebido desempenham um papel fundamental.

2. Desempenho auto-assitido: à medida que a experiência é assimilada, o aprendiz torna-se capaz de corrigir a si próprio e buscar o aprimoramento de maneira autônoma. Embora a tarefa ainda não seja completamente dominada, ela pode ser realizada com certo esforço.

3. Internalização, automatização, fossilização: com a maturação do desempenho, o desenvolvimento se completa e deixa de requerer correção e aprimoramento constantes. A partir de então a capacidade é dita "internalizada" e a estagnação pode levar a um distanciamento das necessidades que surgirem ao longo do tempo, chegando à "fossilização".

4. Desautomatização: quando surgirem novos desafios e dificuldades no desempenho da tarefa o sujeito pode precisar do auxílio de terceiros

> ou rever e atualizar suas competências até então desempenhada de maneira automática, retornando então para os estágios iniciais.

Dá-se o nome de zona de desenvolvimento proximal à diferença entre aquilo que o sujeito é capaz de desempenhar com o auxílio de outros (nível de desenvolvimento potencial) e o que ele pode desempenhar sozinho (nível de desenvolvimento real). Diz respeito, portanto, às capacidades do indivíduo que estão em maturação, excluindo aquelas que ele é incapaz de desempenhar no momento ou as que já estão plenamente desenvolvidas. O mecanismo autocorretivo consciente típico do segundo estágio, na aquisição de uma segunda língua, pode se manifestar como o "monitor" de Krashen: quando este se caracteriza como uma manifestação intrapsíquica de regulação do próprio desempenho que recorre a regras gramaticais aprendidas e utilizadas conscientemente.

Outra associação que se tornou comum é que a ZDP seja equivalente à hipótese do *input* compreensível de Krashen. De acordo com Krashen (1981), em geral a aprendizagem de uma língua ocorre efetivamente não quando o estudante recebe um input lingüístico que coincida exatamente com o seu nível "i", mas quando o input está ligeiramente acima de seu estágio de desenvolvimento, mas ainda compreensível (i + 1). Entretanto, essa associação entre os dois conceitos vem sendo recentemente questionada. Thorne (2000), por exemplo, faz a seguinte análise:

> "A ZDP de Vygotsky vem sendo igualada de maneira pouco convincente com a metáfora do *input* compreensível (i + 1) de Krashen. As diferenças entre esses dois conceitos tem sido discutida em outros lugares (de Guerrero 1996; Dunn & Lantolf 1998), mas resumidamente, Krashen desenvolveu a hipótese do *input* - a noção de que a aquisição de uma língua ocorre quando um indivíduo está submetido a *input* na língua-alvo em [nível] i + 1, onde i é o nível de competência atual do aprendiz e + 1 denota o estágio que sucede imediatamente i em uma ordem natural da seqüência de desenvolvimento. O estudante se move de um estágio i para o estágio i + 1 ao compreender *input* contendo i + 1 (Krashen 1982). O conceito de ZDP de Vygotsky, em nítido contraste, envolve o que um indivíduo pode realizar ou desempenhar em colaboração com outro mais competente (ou as pro-

> priedades estruturais do ambiente físico, ou meios e ferramentas mediacionais construídos que podem ter o peso daquilo que é tradicionalmente entendido como atividade 'mental'). Assim i + 1 é uma metáfora para a qualidade do *input* lingüístico e seu efeito na aquisição de línguas, enquanto a ZDP de Vygotsky é uma abordagem teórica para o desenvolvimento baseado numa análise cuidadosa da atividade, possível através da colaboração. O i + 1 de Krashen e a ZDP de Vygotsky, então, não estão relacionados nem em sua conceitualização (um corpo passivo escutando versus atividade colaborativa), alicerces filosóficos (estudante independente versus habilidade pessoal construída através de atividade com outras pessoas e artefatos no ambiente), processos enfatizados (aprendizagem como a de uma criança versus o cumprimento colaborativo de uma tarefa específica), e geralmente, nem em seus resultados (ainda que certamente é possível que um par mais capaz possa prover interação social linguisticamente mediada com o fim de assistir alguém em tarefas e aprendizagem relacionado à língua)." [Aí em cima ele cita (Krashen 1982) e, no meu próprio trabalho, eu não tenho essa referência. Devo acrescentá-la também?]

Embora ainda não seja possível simular a interação entre professor e aluno que caracterizam o estágio inicial da ZDP, à medida que o aluno começa a adquirir um certo grau de auto-suficiência uma ferramenta que promova a resolução de exercícios com *feedback* imediato pode ser utilizada como um auxiliar até alcançar o estágio da internalização (estágio III da divisão proposta por Tharp & Gallimore). Com ela o aluno tem a oportunidade de praticar as competências lingüísticas adquiridas de maneira potencialmente completa e consistente, de acordo com a maturidade do sistema.

Em contrapartida, a supervisão do professor é importante antes e durante a execução dessa atividade. O sistema deve ser adaptado de acordo com o nível do aluno, sugerindo correções que estejam dentro de sua ZDP: nem demasiadamente simples, causando desinteresse no uso da ferramenta, nem avançadas demais, levando o estudante à desmotivação. No caso de uma turma heterogênea, como é o caso de um curso voltado para estudantes de vários países, a transferência lingüística é outro fator fundamental para

a determinação do tipo e nível de detalhamento do *feedback* a ser fornecido. Por exemplo, se a língua sendo estudada é o francês e o aluno produzir a expressão **la pont* (“a ponte”, com o artigo definido feminino, sendo que em francês este substantivo é masculino), o sistema deve saber qual é a língua nativo do estudante. Para um estudante de língua portuguesa, uma curta resposta indicando o problema de concordância de gênero seria suficiente. Para um falante nativo do inglês, pode ser útil uma explicação mais detalhada enfatizando que na língua francesa existe uma convenção de gênero masculino ou feminino para seres inanimados, já que em sua língua original tal distinção não é feita. Deve existir, portanto, flexibilidade suficiente para que o professor possa adaptar a aplicação de acordo com o perfil dos alunos.

Considerando que a aprendizagem, ou seja, o uso do “monitor” é uma condição importante, mas não suficiente na assimilação de outra língua, deve-se procurar promover nos estudantes a atitude de um “usuário ideal do monitor” (Krashen, 1981, p. 37), usando essa prática nas situações onde ela não interfere na comunicação, como na expressão escrita. É recomendável que o sistema mantenha um registro de utilização, seja para fins estatísticos, para acompanhamento individualizado e comparação com o desempenho em classe, para identificar aprimoramentos na ferramenta ou adequá-la melhor ao perfil do educando.

# Capítulo 3

# Correntes teóricas

## 3.1 Empirismo e Racionalismo em PLN

A pesquisa em Lingüística, que até o início da década de 50 era predominantemente baseada na tradição estruturalista iniciada por Saussure e trabalhava com corpus elaborados manualmente ou com recursos computacionais que começavam a ser empregados na época, sofreu uma mudança radical com a publicação de *Syntactic Structures* (Chomsky, 1957). A partir de então, as abordagens derivadas da gramática gerativa rivalizaram com outras abordagens distribucionais, uma rivalidade cuja transposição da Lingüística teórica para suas aplicações computacionais foi motivada não apenas por motivos científicos, mas também pessoais e ideológicos (Pereira, 2000).

Provavelmente essa disputa teve início com um clássico exemplo de Chomsky no qual ele apontava uma suposta deficiência das abordagens distribucionais:

1. Colorless green ideas sleep furiously.

2. Furiously sleep ideas green colorless.

Enquanto a primeira sentença não faz nenhum sentido, mas é considerada sintaticamente bem formada na língua inglesa, a segunda não respeita nenhuma estrutura sintática ou semântica[1]. Uma gramática gerativa

[1]A primeira sentença pode ser traduzida como “Idéias verdes incolores dormem furiosamente”. A segunda é seu inverso, que fere as estruturas sintáticas do inglês possivelmente de maneira mais marcante que sua equivalente em português, sendo esta uma língua mais flexível quanto à disposição das palavras na sentença.

pode capturar essa diferença, mas, para um modelo estatístico da linguagem, ambas as sentenças seriam simplesmente reconhecidas como inválidas, ou seja, com probabilidade zero de ocorrência.

Contudo, Pereira argumenta que "isto se apóia na suposição implícita de que qualquer modelo probabilístico necessariamente atribui probabilidade zero a eventos inéditos. [...]Mas agora nós entendemos que esse método ingênuo ajusta-se demasiadamente aos dados utilizados no treino." (ibid.) Em outras palavras, a critica de Chomsky não considerou os métodos de suavização (*smoothing*) de modelos estatísticos que permitem lidar com a generalização dos dados empiricamente assimilados, os quais desde então vêm obtendo bons resultados nos modelos de linguagem natural (ibid.).

A percepção de Chomsky, somada com outros problemas tais como a falta de poder computacional e de métodos satisfatórios de aprendizagem de máquina, fizeram com que as atenções se voltassem para os métodos simbólicos, ou baseados em conhecimento, até o final da década de 70. No início da década de 80 o grupo da IBM de pesquisa em reconhecimento de voz empregou o poder computacional que havia se tornado disponível com o progresso da tecnologia de hardware e conseguiu desenvolver aplicações usando técnicas probabilísticas com Modelos Ocultos de Markov. Obtiveram resultados promissores, voltando novamente a atenção às abordagens empiricistas, já que as décadas anteriores de experiências com sistemas simbólicos revelaram as limitações desta abordagem, como a complexidade da codificação manual de gramáticas abrangentes e dificuldade de representação de certas propriedades contínuas (não-discretas) da linguagem. Novos algoritmos de aprendizagem de máquina consolidaram esse retorno das teorias estatísticas.

A visão predominante nos últimos anos é a de que ambos os métodos são complementares, cada um com suas particularidades que os tornam mais ou menos pertinentes de acordo com a natureza do problema sendo resolvido (Tsujii, 2000; Balfourier et al., 2002; Frank et al., 2003; Swift et al., 2004). Até o momento, as aplicações de ICALL têm adotado mais intensivamente os métodos baseados em conhecimento, pois estes favorecem as análises profundas que contém informação relevante para a validação sintática e semântica, ao contrário das análises superficiais que geralmente se obtém com técnicas probabilísticas.

Entretanto, página a seguir mostra algo que foi desenvolvido e indica

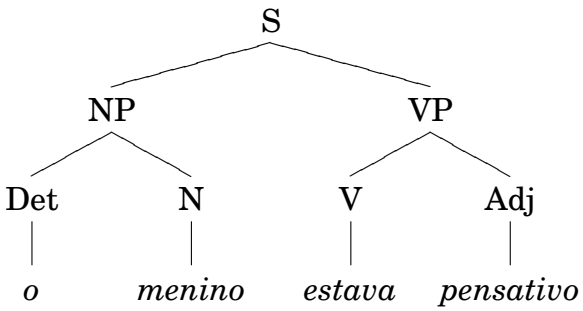


Figura 3.1: Árvore de derivação para “o menino estava pensativo”

que as análises probabilísticas parecem ser viáveis. (http://iit-iti.nrc-cnrc.gc.ca/projects-projets/adverbs_e.html) Mas como é que eu vou citar isso se eu não achei nenhuma informação além dessa página?

## 3.2 Algumas abordagens simbólicas

Embora Chomsky tenha dado origem à teoria das gramáticas formais, suas propostas (Regência e Ligação, Minimalista, etc.) não têm sido muito utilizadas em implementações computacionais, seja pela elevada complexidade, seja por críticas à sua validade lingüística, sendo as “regras transformacionais” um exemplo típico, inclusive em sua formulação mais recente, *Mova-*$\alpha$. Entretanto, suas contribuições teóricas fundamentaram, em maior ou menor grau, todas as teorias e aplicações simbólicas de sintaxe.

Uma das primeiras tentativas de implementação computacional de gramáticas para linguagem natural foram as redes de transição, inicialmente definidas como RTNs (*Recursive Transition Networks*), uma forma simples e eficiente de analisar sentenças. Por exemplo, a sentença *o menino estava pensativo*, cuja árvore de derivação é mostrada na Figura 3.1, poderia ser analisada com as redes de transição mostradas Figura 3.2:

Provavelmente também desejaremos analisar a sentença *a menina estava pensativa*. Entretanto, se acrescentarmos as palavras “a” com categoria “Det”, “menina” com categoria “N” e “pensativa” com categoria “Adj”, a mesma rede de transição vai passar a reconhecer também sentenças como **a menino estava pensativo* e **o menino estava pensativa*, o que não seria apropriado. A solução, usando RTNs, seria criar categorias como “Det-masc”, “Det-fem”, “N-masc”, “N-fem”, “Adj-masc”, “Adj-fem”, “NP-masc”,

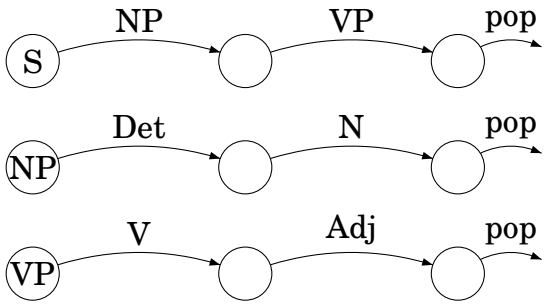

Figura 3.2: Redes de Transição

"NP-fem", "VP-masc", "VP-fem" e alterar as regras para usar as novas categorias. Se formos acrescentar a concordância de número, além da concordância de gênero, a quantidade de categorias seria novamente multiplicada, processo que leva a uma quantidade de tipos inviável.

Também há outro problema: RTNs têm o poder computacional equivalente ao de uma gramática livre de contexto, mas na década de 80 descobriu-se que a linguagem natural não é livre de contexto, mas provavelmente se enquadra num nível intermediário entre as gramáticas livres de contexto e sensíveis ao contexto, chamado de gramáticas ligeiramente sensíveis ao contexto (Marcus et al., 1998).

Para solucionar esses problemas surgiram as ATNs (*Augmented Transition Networks*), que utilizam variáveis chamadas registradores para facilitar a implementação de recursos como concordância e conferem poder computacional superior ao das gramáticas livres de contexto (Woods, 1970).

Embora as ATNs sejam um aprimoramento substancial, certas restrições como sua natureza procedural, a dificuldade para especificar a representação semântica, a falta de recursos que permitam tratar satisfatoriamente línguas que permitem maior liberdade na ordem das palavras, motivaram esforços tanto no aprimoramento das próprias ATNs (Woods, 1980; Prodanof & Ferrari, 1983) quanto a adoção de outras teorias lingüísticas e computacionais que pudessem oferecer alternativas mais adequadas às diversas necessidades em PLN.

Gramática Categorial, Gramática de Cláusulas Definidas, PATR-II, Gramática Léxico-Funcional, TAG, GPSG e HPSG são algumas das teorias que foram implementadas (Cole et al., 1997). Suas implementações têm em comum o uso da *unificação*[2], que vem sendo empregada há décadas em di-

[2]Embora essas gramáticas sejam freqüentemente designadas como "baseadas em uni-

versas áreas de estudo, como prova de teoremas, programação lógica (por exemplo, Prolog) e, evidentemente, PLN (Knight, 1989). A unificação é uma operação na qual dois conjuntos de informação são combinados, dando origem a um único conjunto contendo as informações de ambos os operandos. Entretanto, nem todos os pares de informações são unificáveis, a unificação pode falhar caso os operandos não sejam *compatíveis*. Portanto, a unificação de um $x$ qualquer com um $y$ pode produzir um $xy$ que combina as informações de $x$ e $y$, ou a operação pode falhar, tendo um resultado indefinido representado por $\perp$ (*bottom*).

Cada um desses formalismos gramaticais tem sua própria definição de unificação. Em linhas gerais, a unificação é vista como um mecanismo composicional, onde cada sentença é vista como o resultado da unificação de constituintes maiores, representando sintagmas, e constituintes menores, palavras ou expressões. A unificação entre constituintes falha quando o resultado seria um constituinte agramatical, de forma que a gramática rejeita ou reconhece as sentenças de acordo com a compatibilidade das unidades mais elementares que a constituem.

## 3.3 Head-Driven Phrase Structure Grammar

Kay (1979, apud Knight, 1989) foi o precursor no uso da unificação para processamento sintático de linguagem natural, formalizando a noção lingüística de "traços" em "estruturas de traços". Uma estrutura de traço é um conjunto de atributos e respectivos valores que descrevem uma entidade qualquer.

$$\begin{bmatrix}\text{NOME} & \textit{João}\\ \text{FAIXA-ETÁRIA} & \textit{maior}\end{bmatrix} \sqcup \begin{bmatrix}\text{HABILITADO} & \textit{sim}\\ \text{FAIXA-ETÁRIA} & \textit{maior}\end{bmatrix} = \begin{bmatrix}\text{NOME} & \textit{João}\\ \text{HABILITADO} & \textit{sim}\\ \text{FAIXA-ETÁRIA} & \textit{maior}\end{bmatrix} \quad (3.1)$$

$$\begin{bmatrix}\text{NOME} & \textit{Joãozinho}\\ \text{FAIXA-ETÁRIA} & \textit{menor}\end{bmatrix} \sqcup \begin{bmatrix}\text{HABILITADO} & \textit{sim}\\ \text{FAIXA-ETÁRIA} & \textit{maior}\end{bmatrix} = \perp \quad (3.2)$$

A Equação 3.1 representa uma pessoa chamada *João*, maior de idade, que é aprovada nos exames para obter permissão para digirir. A unificação, aqui indicada pelo operador $\sqcup$, está combinando os traços em *João* (NOME,

ficação", tal expressão é criticada visto que a unificação é apenas uma operação que pode ser utilizada para implementá-las e não um fundamento teórico no qual elas se baseiam. Veja (Pollard, 1996) e também (Pereira, 1993, seção 5.1), onde o assunto é brevemente discutido.

FAIXA-ETÁRIA) com os traços de um condutor habilitado (HABILITADO, FAIXA-ETÁRIA). Nesse caso o traço FAIXA-ETÁRIA existe nos dois operandos, mas como ambos têm o mesmo valor (*maior*) ele aparece normalmente na estrutura resultante.

A Equação 3.2 mostra uma situação análoga, mas neste caso a pessoa em questão é *Joãozinho*, menor de idade. Agora a unificação falha, pois o traço FAIXA-ETÁRIA existe nos dois operandos, porém com valores diferentes. O valor de FAIXA-ETÁRIA explicitamente definido como *maior* na estrutura de traços correspondente a condutores habilitados tem o efeito de impedir que menores de idade obtenham permissão para dirigir, fato que se reflete na falha da unificação.

As estruturas de traços podem ser representadas através de *matrizes de atributo-valor* (MAV), como as que aparecem nas equações 3.1 e 3.2. Uma representação alternativa é na forma de *grafos direcionais*, onde os arcos são os traços e os vértices são os valores correspondentes, como exemplificado na Figura 3.3.

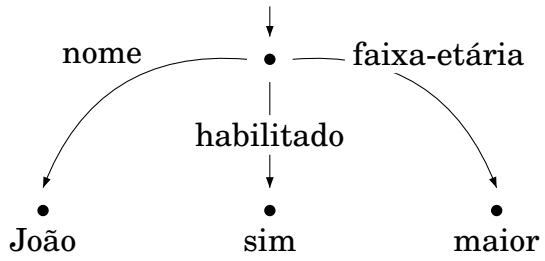


Figura 3.3: Grafo direcional

Em geral as estruturas de traços não são tão simples. O valor de um traço pode ser, por sua vez, outra estrutura de traços. Também é possível uma situação na qual dois ou mais traços tenham a mesma estrutura como valor, não apenas como uma cópia, mas que o destino dos arcos correspondentes aos traços de fato apontando para a mesma estrutura. A essa ocorrência dá-se o nome de *compartilhamento de estruturas*, ou *reentrância*. Nas MAVs as reentrâncias são indicadas por pequenos quadrados contendo um índice, cujo valor e ordem de ocorrência não é importante, apenas indicam que os traços co-indexados fazem referência à mesma estrutura. Como um grafo, a reentrância consiste em dois ou mais arcos apontando para um mesmo vértice.

As categorias de uma gramática de língua natural também podem ser

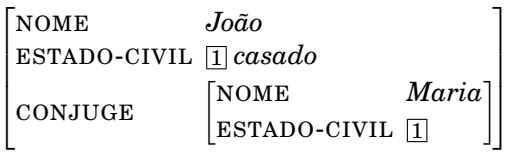


Figura 3.4: Compartilhamento de estruturas

representadas por estruturas de traços. No lugar de simplesmente "V", "Det" e "N" podemos ter categorias complexas:

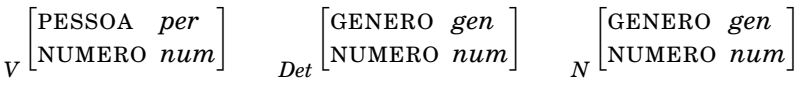


Figura 3.5: Categorias complexas

Com categorias complexas e reentrâncias, podemos representar um constituinte com certa economia na quantidade de tipos e prevendo relações com os demais constituintes, tal como concordância verbal e nominal. Com uma única categoria "Det" contendo um traço para o gênero e outro para o número, pode-se representar todos adjetivos que, com as categorias atômicas das RTNs e outras teorias anteriores, constituiriam quatro categorias distintas (no caso do português), o que se torna inviável para o desenvolvimento de qualquer gramática de abrangência significativa.

A Figura 3.6 mostra a representação composicional de um sintagma nominal ("o menino"), cuja representação fonológica é determinada pela combinação das representações fonológicas dos constituintes que domina. As estruturas dos traços GÊNERO e NÚMERO são compartilhadas entre os constituintes, o que impede a produção de sentenças como **a menino* ou **os menino* — os valores *masc* e *fem* não são compatíveis.

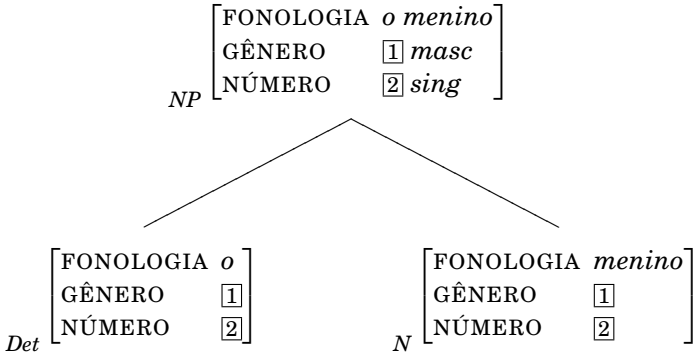


Figura 3.6: Estrutura de traços para "o menino"

Uma das primeiras teorias a utilizar estruturas de traços foi a GPSG (*Generalized Phrase Structure Grammar*). Entretanto, ela tem um poder

computacional equivalente ao de uma gramática livre de contexto (Borsley, 1996, p. 16–19), logo, como foi mencionado na página 27, não constitui um modelo satisfatório para linguagem natural. Essa e outras limitações conduziram à elaboração de uma nova teoria, HPSG (*Head-Driven Phrase Structure Grammar*). Algumas das características da HPSG são:

- Não está limitada a linguagens livres de contexto. Segundo Carpenter (1991), com HPSG é possível gerar linguagens recursivamente enumeráveis, ou seja, correspondentes a gramáticas irrestritas (Tipo 0).

- Foi baseada em inúmeras teorias pré-existentes, incluindo Regência e Ligação, Gramática Categorial, Gramática Léxico-Funcional, GPSG, bem como em pesquisas na área de Ciência da Computação. Tais características contribuiram para que se tornasse uma teoria bastante satisfatória do ponto de vista lingüístico e computacionalmente viável.

- Não faz uso explícito de categorias vazias ($\varepsilon$), nem possui qualquer forma de transformação: fenômenos como dependências de longa distância são tratados com compartilhamento de estruturas. Enquanto na gramática transformacional chomskyana a sentença possui uma Estrutura Profunda, sobre a qual são aplicadas transformações para se obter a Estrutura Superficial, HPSG é uma teoria não-derivacional. De fato, a noção de movimento de constituintes é explicitamente rechaçada, senão como inválida, pelo menos como supérflua (Pollard & Sag, 1996).

- Sua unidade lingüística básica é o *signo*, seguindo a definição quase axiomática de Saussure, uma associação arbitrária entre significante e significado. Todos os constituintes, sejam lexicais ou sintagmáticos, até uma sentença completa, são signos que agregam informações fonológicas, sintáticas e semânticas. As estruturas de traços são a "concretização" dos signos, considerando HPSG como um formalismo gramatical.

- Versões básicas do formalismo (Borsley, 1996, p. 69) definem em cada signo o traço PHON que referencia sua representação fonológica (significante) e o traço SYNSEM para as informações sintáticas e semânticas (significado). Uma alternativa é haver traços separados SYN e SEM. Os signos sintagmáticos têm um traço adicional, DAUGHTERS,

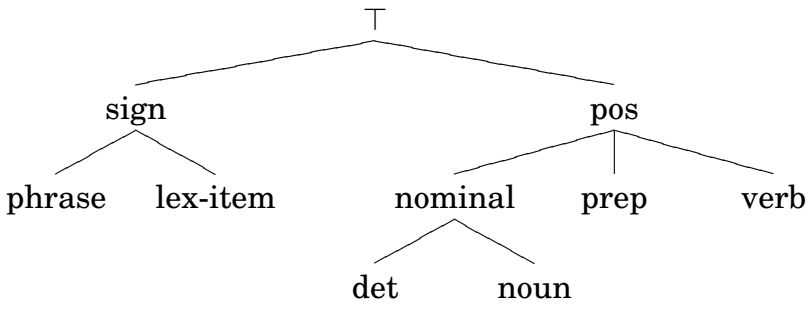


Figura 3.7: Parte de uma hierarquia de tipos

que especifica os signos dominados por este. Sendo as estruturas de traços empregadas consistentemente em todos os níveis de uma sentença, considera-se a HPSG como uma teoria *fractal*, onde as partes menores são análogas às partes maiores que as contêm. Embora árvores sejam uma representação consagrada e conveniente também em HPSG, como as sentenças e seus constituintes são estruturas de traços, as MAVs são descrições mais apropriadas quando a precisão for mais importante que a simplicidade.

- É uma teoria fortemente lexicalizada. Os signos lexicais são relativamente complexos, contendo diversas informações tais como os especificadores e complementos com os quais este item lexical pode se combinar. Essa ênfase lexicalista acarreta uma redução da quantidade de regras sintagmáticas necessárias, redução ainda mais acentuada pela adoção de *princípios universais*, conceito proveniente da Gramática Universal de Chomsky. Um dos princípios mais elementares e importantes é o Princípio do Traço HEAD, formulado como:

  > *O valor* HEAD *de um sintagma com núcleo é idêntico ao valor* HEAD *da filha correspondente ao núcleo.* (Borsley, 1996, p. 50)

As estruturas de traços em HPSG são *estruturas de traços tipadas*, a cada estrutura é atribuído um tipo. Os tipos da gramática fazem parte de uma única *hierarquia de tipos*, como a da Figura 3.7:

Uma gramática pode ter centenas de tipos dispostos hierarquicamente a partir de um elemento mais geral $\top$ (*top*). Os tipos mais específicos herdam as definições dos tipos mais gerais, evitando efeitos indesejados como

a multiplicação da quantidade de regras necessárias e facilitando a codificação do léxico. Em gramáticas lexicalizadas como HPSG este é um recurso importante, já que as entradas lexicais em geral têm bastante informação. Também é permitida a herança múltipla, onde um tipo herda as definições de dois ou mais tipos distintos.

A Figura 3.8 mostra um signo lexical simplificado, correspondente ao verbo "perseguir". Os tipos aparecem subscritos, à esquerda da estrutura à qual se referem, mas freqüentemente são omitidos por simplicidade — nesse caso foram todos explicitamente indicados.

Esta estrutura é definida como verbo transitivo, com sua representação fonológica no traço PHON. Seu núcleo (HEAD) é um verbo, que neste caso faz alusão ao próprio signo (*verbo-trans* é um tipo específico de *verbo*, portanto são compatíveis). O traço MOD como uma lista vazia indica que ele não está associado a nenhum adjunto (se houver algum adjunto, ele será referenciado apenas no nível sintagmático, não no lexical). Tanto seu especificador (SPR) quanto seus complementos (COMPS) são sintagmas que têm substantivos como núcleo.

As informações semânticas estão agrupadas nos traços SEM. O traço RELS compõe as relações semânticas deste signo através do compartilhamento de estruturas: $\boxed{3}$ o evento "perseguir", $\boxed{1}$ o sujeito que persegue, $\boxed{2}$ aquilo que é perseguido.

$$
\textit{verbo-trans}\left[\begin{array}{ll}
\text{PHON} & \textit{perseguir} \\
\text{HEAD} & \textit{verbo}\left[\text{MOD } \langle\,\rangle\right] \\
\text{SPR} & \left\langle \textit{sintagma}\left[\begin{array}{l}\text{HEAD } \textit{substantivo} \\ \text{SEM} \mid \text{INDEX } \boxed{1}_{\textit{indice}}\left[\text{INSTLOC } \textit{instloc}\right]\end{array}\right]\right\rangle \\
\text{COMPS} & \left\langle \textit{sintagma}\left[\begin{array}{l}\text{HEAD } \textit{substantivo} \\ \text{SEM} \mid \text{INDEX } \boxed{2}_{\textit{indice}}\left[\text{INSTLOC } \textit{instloc}\right]\end{array}\right]\right\rangle \\
\text{SEM} & \textit{semantica}\left[\begin{array}{l}\text{INDEX } \boxed{3}_{\textit{evento}}\left[\text{INSTLOC } \textit{instloc}\right] \\ \text{RELS } \langle \boxed{3}, \boxed{1}, \boxed{2} \rangle\end{array}\right]
\end{array}\right]
$$

Figura 3.8: Exemplo de estrutura de traços tipada

## 3.4 Trabalhos anteriores

DAQUI PARA FRENTE AINDA É TUDO RASCUNHO, NEM VALE A PENA LER

Referenciar aqui o GRAMMAR FRAMEWOPKS IN INTELLIGENT CALL, que fala da teoria.

Uma versão adaptada de ATNs foi utilizada para a construção do revisor gramatical ReGra, segundo os autores o mais completo revisor para a língua portuguesa (Rino et al., 2002).

Revisor gramatical desenvolvido por alunos e professores do Instituto de Ciências Matemáticas (ICM) da USP, em São Carlos e utilizado no Word, que segundo os autores é atualmente o mais completo corretor para a língua portuguesa. Falar também do Banzai (Nagata, 1997).

Citar os sistemas anteriores, quais teorias utilizaram (HPSG ou não) e os resultados obtidos. Bom lugar para falar do ICICLE, voltado para usuários da ASL. Explicar que ICALL engloba outra questão lingüística universal, que é o tratamento de sentenças malformadas que tanto existem tanto na linguagem falada quanto escrita e que é um tema importante tanto do ponto de vista teórico quanto das implementações de sistemas. Um sistema real deve ser capaz de lidar com essas situações, indicando o nível de gramaticalidade de sentenças ou mesmo do discurso. E que tal o FreeText? (http://www.latl.unige.ch/freetext/)

Falar do Arboretum, claro!

# Capítulo 4

# Protótipo desenvolvido

## 4.1 Formalismo gramatical

Falar das “quatro estratégias de desenvolvimento de gramáticas”, citando que nesse projeto foi utilizada uma gramática pré-existente.

Na Figura 3.4, página 30, foi apresentado um grafo direcionado, mas o que rola na verdade é um *grafo direcional acíclico*, não que a teoria não permita mas não é realmente necessário e a ausência permite simplificar o analisador.

Falar que foi baseado no LKB, citando as similaridades e esmiuçando as diferenças. Colocar aqui as formalizações algébricas apropriadas, que já estão descritas no material a respeito do LKB. Falar do que foi implementado e o que não foi - particularmente não implementei MRS (semântica). Como são implementadas a negação e a disjunção na hierarquia de tipos.

## 4.2 Técnicas de correção

Fazer uma pesquisa sobre as técnicas existentes, por exemplo, regras de malformação (a que deverá ser usada aqui), flexibilização de restrições e outras. Discorrer brevemente sobre as características de cada uma. Falar aqui também da correção de pontuação, que é importante, mas não será implementada, fazendo referência aos artigos acadêmicos sobre o assunto. Detalhar especificamente como a correção foi implementada neste sistema.

Também vou falar - não sei se aqui ou em uma nova seção - dos efeitos da subcorreção (falsos positivos) e supercorreção (falsos negativos) na aprendizagem, enfatizando que a hipercorreção deve ser evitada o tanto quanto possível, pois seus malefícios só podem ser contornados pelos alunos muito

avançados, para os quais o sistema pode mostrar-se não compensador. Um gráfico com uma intersecção pode cair bem aqui.

Learners learn better when they must answer questions rather than simply read material and when they receive feedback on their responses (Pany and McCoy, 1998; Sassenrath, 1975). Learners provided with feedback outperform thise given minimal or no feedback (Van Dusen and Worther, 1995, p. 30; as cited in Gregiore et. al.). Learners who received explicit feedback were shown to perform better than those who were given implicit feedback (Caroll & Swain 1993)

## 4.3 Implementação

Foi feito em Java, utilizando-se de alguns arquivos do LKB (por isso o ProcessadorLisp - que não é um interpretador Lisp, serve apenas para esse caso). Mencionar as inúmeras otimizações, fazendo referência à literatura de onde elas foram tiradas. Citar as otimizações importantes que não foram implementadas (recorrer à álgebra aqui também, se necessário). Falar do bottom-up chart parser, do livro de onde foi tirado, e como serviu para a gramática livre de contexto que implementei antes da HPSG. Falar (aqui?) qual foi a gramática utilizada como exemplo, e citar a função especial que eu criarei para carregar as regras de má-formação, marcando-as como tais.

Ela tem: chart parser bottom-up, GLC, HPSG, TDL, Lisp, LKB, mal.

# Capítulo 5

# Resultados obtidos

Ainda não sei o que colocarei aqui, talvez o trabalho que deu para elaborar as regras e o resultado obtido a partir delas, para tentar prever o esforço necessário para a construção de um sistema funcional. Dependendo da precisão dos dados que eu conseguir, poderei criar gráficos e tabelas.

Posso mencionar o custo de desenvolvimento de algumas gramáticas que eu conheço, como a ERG e a do francês.

# Capítulo 6

# Conclusão e perspectiva futura

Achei a seguinte sugestão para conclusão na Internet, mas me questiono se o lugar do "Capítulo 5: Resultados obtidos" também seria aqui: É o último capítulo do projeto e deve apresentar uma análise crítica do trabalho realizado e dos resultados obtidos, inclusive com sugestões para trabalhos futuros. Na conclusão é importante separar a análise que se refere ao projeto (métodos, técnicas, ferramentas e aprendizado) da análise que se refere ao produto (referências críticas aos resultados e novas versões).

O que pode ser feito:

- Otimização (especialmente quanto ao uso da memória e cache do léxico).
- Melhor tratamento dos erros sistemáticos, como na morfologia (plurais irregulares, conjugação de verbos) ou na ortografia (falta de acentos, letras de mesma pronúncia - aí vem os pares mínimos, etc.), sugestão de correção gramatical (semântica). Usar a web pela facilidade de implantação, registro de atividades e hipertexto.
- Mais testes com regras de malformação, experimentação com outras técnicas de correção e/ou aprimoramento da técnica atual usando dados reais de problemas de transferência lingüística. O sistema deve gravar um registro do uso, que pode servir para fins pedagógicos ou de aprimoramento da ferramenta.
- Informação gramatical também sobre sentenças bem formadas, pois pode ser interessante para estudo de sintaxe.
- Incorporação de pontuação na correção sintática.

- Faz sentido pensar em algo apresentado em “Multiword Expressions - A Pain in the Neck for NLP”?

Algumas dicas sobre pontuação: Bad grammar and punctuation really bugs me. These forums are FULL of it. I stumbled on this funny story today from a new book, Eats, shoots and leaves.

A panda walks into a cafe. he orders a sandwich, eats it, then draws a gun and fires two shots in the air. “Why” asks the confused waiter, as the panda makes toward the exit. The panda produces a badly punctuated wildlife manual and tosses it over his shoulder. “I’m a panda”, he says at the door. “Look it up.” The waiter turns to the relevant entry and, sure enough, finds an explanation.

“Panda. Large black-and-white bear-like mammal, native to China. Eats, shoots and leaves.”

Um sistema online que recebe o texto digitado, registra e deixa o usuário corrigir. Os dados coletados servem para:

- acompanhar o desempenho do aluno
- a precisão do sistema
- levantar dados importantes sobre aprendizagem da língua (para isso informações do aluno - usar autenticação - é fundamental)

O sistema deve ter cadastrado o nível e histórico do aluno, erros individuais, erros coletivos, a quantidade de vezes que cada aluno e todo o grupo consultou determinada regra depois de ter sido o erro identificado. Lembrar que mesmo sem regra identificada, sinalizar que há erro é importante! E os alunos podem contribuir ativamente com palavras e regras, além de que cada sentença que deixa de ser gerada ou palavra não reconhecida pode ser automaticamente registrada para evolução do sistema.

Abordagens híbridas podem ser melhores para o usuário final, mas o sistema requer ainda maior conhecimento de processamento de linguagem natural, que desde o princípio não é trivial.

# Apêndice A

# Aqui serão vários anexos

# Referências Bibliográficas


Balfourier, J.-M., Blache, P., & van Rullen, T. (2002). From shallow to deep parsing using constraint satisfaction. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING-02)* Taipei, Taiwan.

Borsley, R. D. (1996). *Modern Phrase Structure Grammar*. Number 11 in Blackwell Textbooks in Linguistics. Blackwell Publishers.

Carpenter, B. (1991). The generative power of Categorial Grammars and Head-Driven Phrase Structure Grammars with lexical rules. *Computational Linguistics*, 17(3), 301–313.

Chomsky, N. A. (1957). *Syntatic Structures*. Cambridge: MIT Press.

Cole, R. A., Mariani, J., Uszkoreit, H., Zaenen, A., & Zue, V. (1997). *Survey of the State of the Art in Human Language Technology*. Cambridge University Press.

Copestake, A. & Flickinger, D. (2000). An open-source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the Second Linguistic Resources and Evaluation Conference* (pp. 591 – 600). Athens, Greece.

Frank, A., Becker, M., Crysmann, B., Kiefer, B., & Schäfer, U. (2003). Integrated shallow and deep parsing: TopP meets HPSG. In *Proceedings of ACL-2003* (pp. 104–111). Sapporo, Japan.

Goyal, P., Mital, M. R., Mukerjkee, A., Raina, A. M., Sharma, D., Shukla, P., & Vikram, K. (2003). Saarthak: A bilingual parser for Hindi, English and code-switching structures. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, EACL* Budapest, Hungary.

Kay, M. (1979). Functional grammar. In *5th Annual Meeting of the Berkley Linguistic Society* Berkley, California.

Knight, K. (1989). Unification: a multidisciplinary survey. *ACM Computing Surveys*, 21(1), 93–124.

Krashen, S. (1981). *Second Language Acquisition and Second Language Learning*. Pergamon.

Lyons, J. (1981). *Lingua(gem) e Lingüística: uma introdução*. Rio de Janeiro: Zahar.

Makino, T., Torisawa, K., & Tsujii, J. (1997). LiLFeS - practical programming language for typed feature structures. In *Proceedings of Natural Language Pacific Rim Symposium*.

Marcus, S., Păun, G., & Martín-Vide, C. (1998). Contextual grammars as generative models of natural languages. *Computational Linguistics*, 24(2), 245–274.

Mayher, J., Lester, N., & Pradl, G. (1983). *Learning to Write, Writing to Learn*. Boynton/Cook Heinemann.

McCoy, K. F. & Masterman, L. N. (1997). A tutor for teaching English as a second language for deaf users of American Sign Language. In *Proceedings of Natural Language Processing for Communication Aids, an ACL/EACL'97 Workshop* Madrid, Spain.

Pereira, F. (1993). Review of "the logic of typed feature structures" by Bob Carpenter. *Computational Linguistics*, 19(3), 544–552.

Pereira, F. (2000). Formal grammar and information theory: Together again? *Philosophical Transactions of the Royal Society*, (pp. 1239–1253).

Pollard, C. (1996). The nature of constraint-based grammar. In *Proceedings of the 10th Pacific Asia Conference on Language, Information and Computation (PACLING'96)* Seoul, Korea: Kyung Hee University.

Pollard, C. & Sag, I. A. (1996). HPSG: Background and basics. The first two sections of this paper are adapted from C. Pollard and I. A. Sag. 1994, Chapter 1, with updates to HPSG-III.

Prodanof, I. & Ferrari, G. (1983). Extended access to the left context in an ATN parser. In *Proceedings of the first conference on European Chapter of the Association for Computational Linguistics* (pp. 58–65).: Association for Computational Linguistics.

Rino, L. H. M., di Felippo, A., Pinheiro, G. M., Martins, R. T., Fillié, V., Hasegawa, R., & das Graças Volpe Nunes, M. (2002). Aspectos da construção de um revisor gramatical automático para o português. *Estudos Lingüísticos*, 1(1), 1–6.

Swift, M., Allen, J., & Gildea, D. (2004). Skeletons in the parser: Using a shallow parser to improve deep parsing. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING-04)* Geneva, Switzerland.

Tharp, R. G. & Gallimore, R. (1988). *Rousing minds to life: Teaching, learning and schooling in a social context*. New York: Cambridge University Press.

Thorne, S. L. (2000). Second language acquisition theory and the truth(s) about relativity. In J. Lantolf (Ed.), *Sociocultural Theory and Second Language Learning* (pp. 219–243). Oxford: Oxford University Press.

Tsujii, J. (2000). Generic NLP technologies: Language, knowledge and information extraction. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics* (pp. 12–22).

Underwood, J. (1984). *Linguistics, Computers, and the Language Teacher: A Communicative Approach*. Rowley, MA: Newbury House.

Warschauer, M. (1996). Computer-assisted language learning: An introduction. In S. Fotos (Ed.), *Multimedia Language Teaching* (pp. 3–20). Tokyo: Logos International.

Warschauer, M. & Healey, D. (1998). Computers and language learning: An overview. *Language Teaching*, (pp. 57–71).

Woods, W. A. (1970). Transition network grammars for natural language analysis. *Communications of the ACM*, 13(10), 591–606.

Woods, W. A. (1980). Cascaded ATN grammars. *Computational Linguistics*, 6(1), 1–12.